Refine Search

Search Results -

Terms	Documents
L20 and L12	2

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L21

Refine Search

Recall Text | Clear | Interrupt

Search History

DATE: Friday, August 20, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L21	L20 and l12	2	L21
L20	L19 same interleav$	5	L20
L19	L18 same l14	21	L19
L18	four near3 l15	31	L18
L17	L15 same l14	208	L17
L16	L15 and l14	226	L16
L15	oc-48	570	L15
L14	oc-192	385	L14
L13	L12 same row same l11	4	L13
L12	fec or (forward adj1 error adj1 correcti$)	8969	L12
L11	sonet or sdh	7046	L11
L10	L2 same sdh	9	L10
L9	L8 same l6	1	L9
L8	number near3 error	29305	L8

L7	L6 and sonet	1	L7
L6	L4 same word	33	L6
L5	L4 same iterat$	2	L5
L4	L2 same location	288	L4
L3	L2 same location same permutation	1	L3
L2	error near3 insert$	4426	L2
L1	error adj1 injection adj1 circuit	15	L1

END OF SEARCH HISTORY

L21: Entry 1 of 2 File: USPT Jun 15, 2004

DOCUMENT-IDENTIFIER: US 6751743 B1
TITLE: Method and apparatus for selecting a first clock and second clock for first and second devices respectively from an up-converted clock and an aligned clock for synchronization

Detailed Description Text (22):
FIG. 6 illustrates a block diagram of portions of protocol processor 520 that includes a framer 600. Deserializer 650 produces a parallel signal 610 from an incoming SONET signal 601 (e.g., an OC-192 data stream) from line side optical receivers (not shown). Receive module 605 processes the parallel signal 610, optionally processes the forward error correction (FEC) information and de-interleaves the OC-192 signal into four OC-48 line rate signals 615 for delivery to downstream OC-48 processors. Transmit module 620 processes four incoming OC-48 system rate signals from the OC-48 processors (signals 625), optionally inserts forward error correction information, and interleaves the four OC-48 signals into an OC-192 signal 630 for transmission by line side optical transmitters (not shown). A CPU Interface module 635 provides the CPU connection to the internal device registers.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

End of Result Set

Generate Collection Print

L21: Entry 2 of 2 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6684350 B1
TITLE: Repetitive pattern testing circuit for AC-coupled systems

Detailed Description Text (23):
FIG. 6 illustrates a block diagram of portions of protocol processor 520 that includes a framer 600. Deserializer 650 converts a serial incoming SONET signal 610 (e.g., an OC-192 data stream) from line side optical receivers (not shown) into a parallel bitstream 660 which is received by receive module 605. Receive module 605 optionally processes the forward error correction (FEC) information and de-interleaves the OC-192 signal into four OC-48 line rate signals 615 for delivery to downstream OC-48 processors. Transmit module 620 processes four incoming OC-48 system rate signals from the OC-48 processors (signals 625), optionally inserts forward error correction information, and interleaves the four OC-48 signals into an OC-192 signal 630 for transmission by line side optical transmitters (not shown). A CPU Interface module 635 provides the CPU connection to the internal device registers.

Previous Doc Next Doc Go to Doc#

L13: Entry 2 of 4 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6683855 B1
TITLE: Forward error correction for high speed optical transmission systems

Abstract Text (1):
Memory requirements and processing delays associated with the application of forward error correction in high speed optical transmissions are substantially reduced by mapping a forward error correction code on a row-by-row basis into unused overhead bytes in a high bit rate signal frame. By applying the forward error correction code to an entire row of the signal frame on a row by row basis, approximately one row needs to be stored at a time thereby reducing the total memory requirements for forward error correction processing. Using SONET as an exemplary application, approximately 1/9th of the entire SONET frame (e.g., one of nine rows) needs to be buffered for forward error correction processing. In an illustrative embodiment, four forward error correction (FEC) blocks are provided for each row for a total of 36 FEC blocks for a frame. Each FEC block comprises four bytes of correction bits for a total of 32 correction bits. These 32 correction bits are mapped to unused overhead and are used for correcting errors in one block of one row of a signal frame, wherein one block covers 1/4th of the row. Other unused overhead bytes in the row can be used to carry error detection codes for detecting multiple errors in a row to determine when forward error correction should be disabled. For example, if a single bit error correcting code is employed, then error correction can be disabled to avoid false corrections if more than one error is detected.

Brief Summary Text (7):
In another proposed scheme, a forward error correction code is applied over three rows of a SONET STS-3 signal (e.g., one-third of a frame). Although this approach requires less overhead as compared to the previous approach, this scheme still requires buffering for one-third of a frame with an associated delay of approximately 45 .mu.s. Furthermore, this scheme also requires multiple forward error correcting algorithms for processing the 3-row block. Scaling to higher rates is also problematic for the same reasons previously set forth. For example, applying this error correction scheme to a STS-192 signal would require parallel processing of at least 64 forward error correction algorithms and buffering for approximately 414,720 bits.

Brief Summary Text (10):
Memory requirements and processing delays associated with the application of forward error correction in high speed optical transmissions are substantially reduced according to the principles of the invention by mapping a forward error correction code on a row-by-row basis into unused overhead bytes in a high bit rate signal frame. By applying the forward error correction code to an entire row of the signal frame on a row by row basis, approximately one row needs to be stored at a time thereby reducing the total memory requirements for forward error correction processing. Using SONET as an exemplary application, approximately 1/9th of the entire SONET frame (e.g., one of nine rows) needs to be buffered for forward error correction processing as compared with an entire SONET frame or one-third of a SONET frame as in prior arrangements. Less memory translates to lower cost and less complexity for implementing forward error correction. Furthermore, delay is reduced as a result of the reduced buffering requirements and processing.

Brief Summary Text (12):
Using a row-by-row mapping approach yields a significant reduction in delay and reduces the amount of memory needed for error correction processing as compared with the prior arrangements. For example, because only one row needs to be buffered and processed at a time, delay for a 10 Gb/s STS-192 signal can be reduced to approximately 15 .mu.s is or less (approximately 1/9th of a 125 .mu.s frame) and memory can be reduced to approximately 17K bytes, both of which are significant improvements over existing forward error correction schemes contemplated for SONET/SDH systems. Moreover, the mapping scheme according to the principles of the invention uses significantly less overhead and does not require parallel processing of a large number of forward error correction algorithms as in prior arrangements. Advantageously, the mapping scheme maintains the structure of the signal and is suitable for various payload rates, e.g., STS-3/3c, STS-12/12c, STS-48/48c, STS-192/192c, as well as higher rate signals that may be used in future applications.

Detailed Description Text (6):
FIG. 2 is a simplified block diagram showing one illustrative embodiment of the forward error correction mapping scheme as applied to SONET STS-192 signal frame 100. In general, forward error correction is applied to signal frame 100 by mapping forward error correction bytes into transport overhead 101 on a row-by-row basis as follows. Four (4) forward error correction (FEC) blocks 205 are provided for each row for a total of 36 FEC blocks for frame 100. FEC blocks 205 are represented as FEC.sub.XY, where X represents the row number (shown as 1-9) and Y represents the block number within the row (shown as A, B, C, and D). Illustratively, FEC 1A-1D represents FEC blocks for row 1, FEC 2A-2D for row 2, and so on. As shown, FEC blocks 205 in a particular row apply to the immediately preceding row, i.e., FEC blocks 1A through 1D cover row 9 of the previous frame, FEC blocks 2A through 2D cover row 1 of the current frame, and so on. The location of FEC blocks 205 within transport overhead 101 will be described in more detail below.

Detailed Description Text (16):
As shown in FIG. 2, FEC bytes are placed in unused section overhead for rows 1-3 and in unused line overhead for rows 5-9 of STS-192 frame 100. For completeness, the detailed placements of FEC bytes in these rows will also be described with reference to the equivalent rate SDH signal, i.e., 10 Gb/s STM-64 signal. The format of an SDH STM-64 frame is well-known and well-documented in SDH-related standards. As such, references to specific mapping locations in both SONET and SDH formats will be understood by those skilled in the art.

Detailed Description Text (17):
In particular, FEC bytes 205 corresponding to block A in rows 1-3 and 5-9 (i.e., FEC 1A, 2A, 3A, 5A, 6A, 7A, 8A, 9A) are mapped to columns 449 through 452 (384+65 through 384+68) and to column 465 (384+81). For the SDH equivalent STM-64 signal, these FEC bytes are mapped to locations S (1-3 and 5-9, 8, 1-4 and 17) where S (x, y, z) is defined with x=row (1-9), y=multi-column (1-9), and z=depth (1-64). Similarly, FEC bytes 205 corresponding to block B in rows 1-3 and 5-9 (i.e., FEC 1B, 2B, 3B, 5B, 6B, 7B, 8B, 9B) are mapped to columns 453 through 456 (384+69 through 384+72) and to column 466 (384+82). For the SDH equivalent STM-64 signal, these FEC bytes are mapped to locations S (1-3 and 5-9, 8, 5-8 and 18). FEC bytes 205 corresponding to block C in rows 1-3 and 5-9 (i.e., FEC 1C, 2C, 3C, 5C, 6C, 7C, 8C, 9C) are mapped to columns 457 through 460 (384+73 through 384+76) and to column 467 (384+83). For the SDH equivalent STM-64 signal, these FEC bytes are mapped to locations S (1-3 and 5-9, 8, 9-12 and 19). Finally, FEC bytes 205 corresponding to block D in rows 1-3 and 5-9 (i.e., FEC 1D, 2D, 3D, 5D, 6D, 7D, 8D, 9D) are mapped to columns 461 through 464 (384+77 through 384+80) and to column 468 (384+84). For the SDH equivalent STM-64 signal, these FEC bytes are mapped to locations S (1-3 and 5-9, 8, 13-16 and 20).

Detailed Description Text (18):

A FEC start byte 400 (also referred to as a FEC correction control byte) can be used to control the enabling/disabling of forward error correction for all 36 blocks within the SONET frame. In one exemplary mapping scheme, FEC start byte 400 can be mapped to row 1, column 469 (384+85) of the STS-192 frame or in the equivalent STM-64 position S (1, 8, 21). As shown in FIG. 4, an exemplary byte definition for FEC start byte 400 comprises eight (8) activity bits 401. In operation, FEC start byte 400 can be beneficially used if an application requires corrections to be enabled or disabled in-service without causing hits on traffic.

Detailed Description Text (21):
Referring again to FIG. 2, FEC bytes 205 corresponding to blocks A through D in row 4 (i.e., FEC 4A, 4B, 4C, 4D) are mapped to the S0 bit location of the last 160 H1 bytes in the line overhead corresponding to columns 33-192 of STS-192 frame 100 or, in the equivalent SDH signal format, positions S (4, 1, 33) through S (4, 3, 64). More specifically, FEC bytes corresponding to block A in row 4 (i.e., FEC 4A) are mapped to columns 33-64 and 161-168, FEC bytes corresponding to FEC 4B are mapped to columns 65-96 and 169-176, FEC bytes corresponding to FEC 4C are mapped to columns 97-128 and 177-184, and FEC bytes corresponding to FEC 4D are mapped to columns 129-160 and 185-192. It should be noted that, for SDH applications, the SS bit mismatch may need to be disabled for STM-64 interfaces in accordance with applicable standards.

Detailed Description Text (30):
Referring to FIG. 8, the FEC decoding functions are shown relative to the receive functions of the system. In operation, SONET/SDH framing (block 801) and SONET/SDH descrambling (block 802) are carried out using well-known processing techniques in accordance with SONET/SDH-related standards. After descrambling, the FEC start byte is located (block 803) and, depending on the value of the start byte, forward error correction will either be enabled to allow corrections or disabled as shown. FEC blocks for row 1, i.e., FEC 1A through 1D, are exclusively OR'd (XOR) with a switchable fixed pattern as shown in block 804 and as described above. The FEC BIP-8 parity byte is then calculated (block 805) over its corresponding block of data. As previously described, FEC BIP-8 parity can be used to detect multiple errors and to appropriately enable or disable forward error correction depending on whether the number of errors detected is within the correction capability of the selected forward error correction algorithm.

Detailed Description Text (31):
After data is stored (block 806), FEC correction bits are then calculated (block 807) to determine the location of any errors. Correction of bit errors can then take place as shown in block 808 assuming that error correction has not been disabled. Upon completion of error correction, the B1 byte is then calculated and compared (block 809) using well-known SONET/SDH processing techniques. The Z0 bytes and S0 bits are then rearranged as shown in blocks 810 and 811, respectively, by copying the contents back into their respective rows. After the S0 bits are rearranged, B2 byte compensation is carried out (block 812) due to the S0 rearrangements. For example, the B2 byte requires compensation for removing the FEC correction bits from row 4. Other section overhead receive byte processing (block 813), B2 Byte calculation (block 814), and line overhead receive byte processing (block 815) is then carried out using well-known SONET/SDH processing techniques.

Detailed Description Text (36):
As previously indicated, forward error correction can be applied to an entire SONET frame so that errors can be corrected in both the transport overhead and payload, with some exceptions. In general, the FEC blocks should not cover the FEC start byte, the FEC BIP-8 bytes, section overhead in rows 1-3, the B2 bytes, and the rearranged Z0 bytes and S0 bits. For example, the FEC start byte may change values during transmission as a result of a facility defect, for example. As such, applying FEC to the start byte may result in a false correction.

Detailed Description Text (44):
According to another aspect of the invention, a provisionable forward error correction capability is provided in which forward error correction processing can occur at the SONET section layer or the SONET line layer depending on the application. For example, forward error correction processing at the line layer may be advantageous when overall network delay is the most significant consideration. For example, when processing at the line layer according to the principles of the invention, a pass-through function is employed at regenerators whereby forward error correction bytes in the first three rows of a SONET frame are passed through. Conversely, forward error correction processing at the section layer may be advantageous when overall network delay is not as important as achieving the lowest possible bit error rate.

Detailed Description Text (45):
The embodiments shown and described herein for mapping forward error correction on a row-by-row basis may be incorporated in devices and systems using techniques well known to one skilled in the art. For example, the forward error correction mapping scheme may be embodied in an application specific integrated circuit (ASIC) using VLSI technology for a SONET or SDH terminal network element.

Detailed Description Text (46):
It should be understood that the particular embodiments and applications described above are only illustrative of the principles of the invention. Those skilled in the art may devise other suitable implementations, without departing from the spirit and scope of the teachings herein, for other signal structures and rates. For example, the principles of the invention may be employed for SONET and SDH signal rates other than STS-192/STM-64 by modifying mapping locations for the various FEC correction bits and bit interleaved parity bits consistent with the teachings herein. Furthermore, the principles of the invention may be applied to any type of signal structure that is conducive to a row-by-row mapping of forward error correction to improve bit error rate performance while minimizing memory requirements and associated delay. Accordingly, the scope of the invention is limited only by the claims that follow.

Previous Doc Next Doc Go to Doc#

L1: Entry 4 of 15 File: USPT Nov 8, 1994

DOCUMENT-IDENTIFIER: US 5363379 A
TITLE: FDDI network test adaptor error injection circuit

Brief Summary Text (3):
This invention relates to a circuit for injecting data containing error into an electronic circuit for which data containing errors may not normally be transmitted. In particular, an error injection circuit for testing a fiber optic distributed data interface (FDDI) network is described.

Brief Summary Text (16):
These objects, and other advantages to become apparent, are achieved by the error injection circuit (EIC) of this invention. The EIC is situated between the FORMAC and ENDEC logic, and allows the transmission of invalid frames of data, valid frames, and invalid/valid line state symbols streams onto a fiber optic bus. The EIC employs a shadow RAM concept to allow it to operationally coexist with the FORMAC, thereby providing a single test adapter design to meet the requirements of a wide range of FDDI hardware/software test environments.

Detailed Description Text (6):
The Error Injection Circuit (EIC) 28 is located between the FORMAC 20 and ENDEC 30. When a FDDI card is in the error injection mode the data to be transmitted originates from the EIC 28. In normal transmission the data comes from the FORMAC 20. In order to provide this interface, the EIC 28 must monitor various status/handshake signals provided by the FORMAC and DPC, and must drive additional control bits to the EIR 24 and multiplexer (MUX) 26. The Error Injection Logic (EIL) 32 provides this capability, as further described below.

L1: Entry 7 of 15 File: USPT Mar 19, 1991

DOCUMENT-IDENTIFIER: US 5001712 A
TITLE: Diagnostic error injection for a synchronous bus system

Abstract Text (1):
Bus error injection circuit generates bus errors to test proper operation of bus error detection and recovery in a system of modules interconnected by a synchronous digital bus. Application of the circuit is bus error detection and recovery tests for a physical realization of the system. The bus error injection circuit can be replicated on a number of modules interconnected by a synchronous bus to provide multiple sources of error injection. One module, or multiple modules, with error injection circuitry is designated as the source(s) to inject a transient bus error. The bus error injection circuitry monitors the bus to determine when the module is a participant in a bus transfer cycle on the bus. An error injection counter decrements for each such cycle. When the counter output value is one, the module derives its error injection pattern onto bus signal lines in place of the signal line values normally generated. When the counter output reaches zero, the count enable is disabled and the signals normally supplied for the next bus cycle are enabled to the bus.

Brief Summary Text (9):
A diagnostic error injection circuit is utilized for a number of modules in a digital system of modules interconnected by a synchronous digital bus. One module (or more) is selected to inject errors onto the synchronous bus during the transfer of a bus message. A module that receives bus signal lines during the bus cycle in which the bus error(s) is injected detects the bus error(s) and recovers from the error(s), as specified by the particular bus standard that is implemented, thereby verifying proper operation of the bus error detection and error handling logic. The error injection circuit provides two modes of operation. During the normal, operational mode, bus signals are driven by a module in accordance with the protocol defined by a particular bus specification.

Detailed Description Text (8):
In the present invention, error injection tests error detection and recovery logic included in functional hardware that uses the erroneous set of signals as inputs. Diagnostic software can be used to initialize the error injection parameters before the hardware test is run and to check hardware conditions set by the normal hardware error detection and recovery logic (for example, faults, interrupts, branch conditions, etc.) after the hardware test is run. However, there is no diagnostic software intervention during the test of a particular error condition. The error injection circuit serves as the mechanism by which the error condition is produced, and is completely separate from the error detection and recovery circuits.

L1: Entry 3 of 15 File: USPT Feb 16, 1999

DOCUMENT-IDENTIFIER: US 5872910 A
TITLE: Parity-error injection system for an instruction processor

Detailed Description Text (43):
The parity error injection circuit operates in a manner similar to that described above for constant mode, which is selected by scanning high data into both control latches A 304 and B 306.

CLAIMS:

10. For use in a data processing system having an instruction processor for executing a predetermined set of machine instructions, and having a storage device for storing predetermined ones of the machine instructions and for providing ones of the stored machine instructions to the instruction processor, and further having parity error detection circuits for detecting a parity error associated with a corrupted one of the machine instructions that has been provided by the storage system to the instruction processor, and further having re-fetch logic circuits for discarding the corrupted one of the machine instructions and for re-fetching it from the storage system, the parity error injection system, comprising:

a programmable storage device to store selector signals indicative of selected ones of the stored machine instructions; and

error injection circuits coupled to the storage system, to said programmable storage device, and to the parity error detection circuits, to inject parity errors in said selected ones of the stored machine instructions after said selected ones of the stored machine instructions are provided to the instruction processor, and to provide the corrupted machine instructions to the parity error detection circuits.

12. The parity error injection system of claim 10, and further including mode selection logic circuits for selectively enabling said error injection circuits.

13. The parity error injection system of claim 12 wherein said mode selection logic circuits include logic for selectively enabling said error injection circuits when the re-fetch logic circuits are activated.

L7: Entry 1 of 1 File: USPT Nov 12, 1996

DOCUMENT-IDENTIFIER: US 5574717 A
TITLE: Line terminating equipment in SDH networks, using forward error correcting codes

Brief Summary Text (8):
Line rate increases can be avoided when check bits can be mapped into existing unused bytes in a signal form. Grover and Moore proposed an STS-1 path (52 Mbit/s) that is encoded in a (6208,6195) cyclic Hamming code. The proposal is described by the paper entitled "Design and Characterization of an Error-Correcting Code for the SONET STS-1 Tributary" provided by W. D. Grover and T. E. Moore for IEEE Transactions on Communications, Vol. 38, No. 4, p. 467. Thirteen check bits are mapped into Path Overhead (POH) auxiliary bytes in a SONET format. It requires no modification to physical interface or section termination circuits on the line. However, it is not straightforward to apply this proposal to paths other than STS-1. Concatenated Virtual Containers, such as VC-4-Xc (where X=1, 4, 16) will be introduced soon to convey ATM cells, where `ATM` is an abbreviation for `Asynchronous Transfer Mode`. Different codes have to be devised for those high-speed paths. Suzuki proposes an error correcting Hamming code for a concatenation path. The proposal is described by the paper of Japanese Patent Laid-Open No.6-29956, entitled "An Insertion Processing Method of Error Correcting Code in a SDH signal and an Optical Transmission Device", which is invented by Teruhiko Suzuki of Fujitsu Corporation Ltd. of Japan. Herein, a code-word is `VC-4-16c`. According to the proposal, check bytes generated are inserted into a staff area existing for idle bits, wherein the staff area is provided at location between POH and payload in the VC-4-16c. This situation is depicted in FIG. 21. However, the method cannot be accepted for error correction scheme on the path other than VC-4-16c. Different FEC codes are necessary for different path speeds, moreover the error correction scheme is not applicable to a VC-4 frame because it does not have the staff area. Both of the methods are designed under concept that FEC is performed at the path layer. In considering a fact that a transmission line is switched to a protection line at LT-MUX based on bit errors detected by embedded B2 bytes in the MSOH, error corrections at the path layer cannot relieve frequency of undesirable bit-losses caused by line switching.

Other Reference Publication (5):
W. D. Grover et al., "Design and Characterization of an Error-Correcting Code for the SONET STS-1 Tributary," IEEE Transactions on Communications, vol. 38, No. 4, Apr. 1990, pp. 467-477.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L10: Entry 3 of 9 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6683855 B1
TITLE: Forward error correction for high speed optical transmission systems

Brief Summary Text (5):
Forward error correction (FEC) has been in use for many years to improve channel reliability in communication systems. However, applying forward error correction in SONET/SDH systems presents many challenges. In particular, inserting a forward error correction code into SONET/SDH transmission frames is very difficult given the limited amount of unused overhead. Furthermore, forward error correction schemes require a substantial amount of memory for buffering large amounts of data at the receiver in a SONET/SDH system. Delays associated with processing forward error correction in a SONET/SDH system are also a problem, especially for higher rate systems.

Previous Doc Next Doc Go to Doc#